

QSS









Ender's Game
a novel by
Orson Scott Card

c/o Barbara Bova Literary Agency
207 Sedgwick Rd.
West Hartford, CN 06107

Bill —
This is small return
for your very kind gift — but I hope ya'll enjoy getting
"all but this are with publisher or agent). Some spots will be familiar,
but I think a master of computers like you will appreciate what's been added.
And again, thank you for ACTION, BASIC XL, & MAC 65;
BASIC XL has already sped up half our games; I wrote a game my
son has wanted me to write for 6 mos — in 3 hrs., using BASIC XL!
Is there a BASIC XL users group? (or, for that matter, an ACTION
(like me) to start me? (or, for that matter, an ACTION
users group?)
Hope you get a tenth as much
pleasure from this book as I have already
gotten from your kind gifts!
Best,
OSC



Craftsmen Need Precision Tools . . . Programmers! Demand Precision Software!

BASIC XL has twice the speed and twice the power of Atari® BASIC. And yet, as befits a fine craftsman's tool, BASIC XL is even easier to use and more dependable, while including such outstanding major additions as structured programming, string arrays, programming aids, enhanced graphics, and business capabilities.

Atari BASIC is a good starting point. We should know. We wrote it in 1978. Buy BASIC XL. Take advantage of five more years of experience!

So, prepare yourself for some exploration into imaginative programming with BASIC XL! Cartridge, excellent tutorial, reference manual . . . \$99.

Functional, honest, and beautiful describe the simple lines of a craftsman's tools. For the jeweler these tools are an extension of the human hand to better execute complex designs. For you, the Atari programmer, Precision Software tools keep complications out of your programming while allowing you to produce intricate programs.



See the complete collection of OSS Precision Software™ tools!

MAC/65: The fastest 6502 macro assembler/editor package on cartridge . . . \$99.

BUG/65: A powerful debugger. On disk, with OS/A+ . . . \$35.

C/65: The first native mode "small c" compiler for Atari computers. On disk . . . \$80.

ACTION!: The fastest, small computer language ever. A feature-packed cartridge at only . . . \$99.

All products on disk include OS/A+ and also require 48 K.

SEE YOUR LOCAL DEALER!

Call or write for informative brochures.



PRECISION SOFTWARE
A DIVISION OF O.S.S., INC.

1173D S. Saratoga/Sunnyvale Road
San Jose, CA 95129 • (408) 446-3099

ATARI is a trademark of Atari, Inc.
MAC/65, BUG/65, C/65, BASIC XL, OS/A+, and
OSS PRECISION SOFTWARE™ are trademarks
of O.S.S. INC. ACTION! is a trademark of Action Computer
Products.



Craftsmen Need Precision Tools . . . Programmers! Demand Precision Software!

At last! A revolutionary new language which dramatically alters the state-of-the-art in programming tools. ACTION!'s incredible speed and remarkable versatility make it the most powerful tool ever for Atari® computers. Since it is 100's of times faster than Basic, you can even use it to write arcade action games.

A straightforward, easy-to-use language with a superb built-in screen editor that allows you to put the "bells and whistles" into your programs. ACTION! comes in a feature-packed cartridge.

See for yourself. Allow your creativity to soar with ACTION! Cartridge, tutorial, reference manual . . . \$99.

T Functional, honest, and beautiful describe the simple lines of a craftsman's tools. For the violin maker these tools are an extension of the human hand to better execute complex designs. For you, the programmer, Precision Software tools keep complications out of your programming while allowing you to produce intricate programs.



See the complete collection of OSS Precision Software™ tools!

MAC/65: The fastest 6502 macro assembler/editor package on cartridge . . . \$99.

BUG/65: A powerful debugger. On disk, with OS/A+ . . . \$35.

C/65: The first native mode "small c" compiler for Atari and Apple® computers. On disk . . . \$80.

BASIC XL: Twice the power and twice the speed of Atari BASIC in an easy-to-use cartridge . . . \$99.

All products on disk include OS/A+ and also require 48 K.

SEE YOUR LOCAL DEALER!

Call or write for informative brochures.

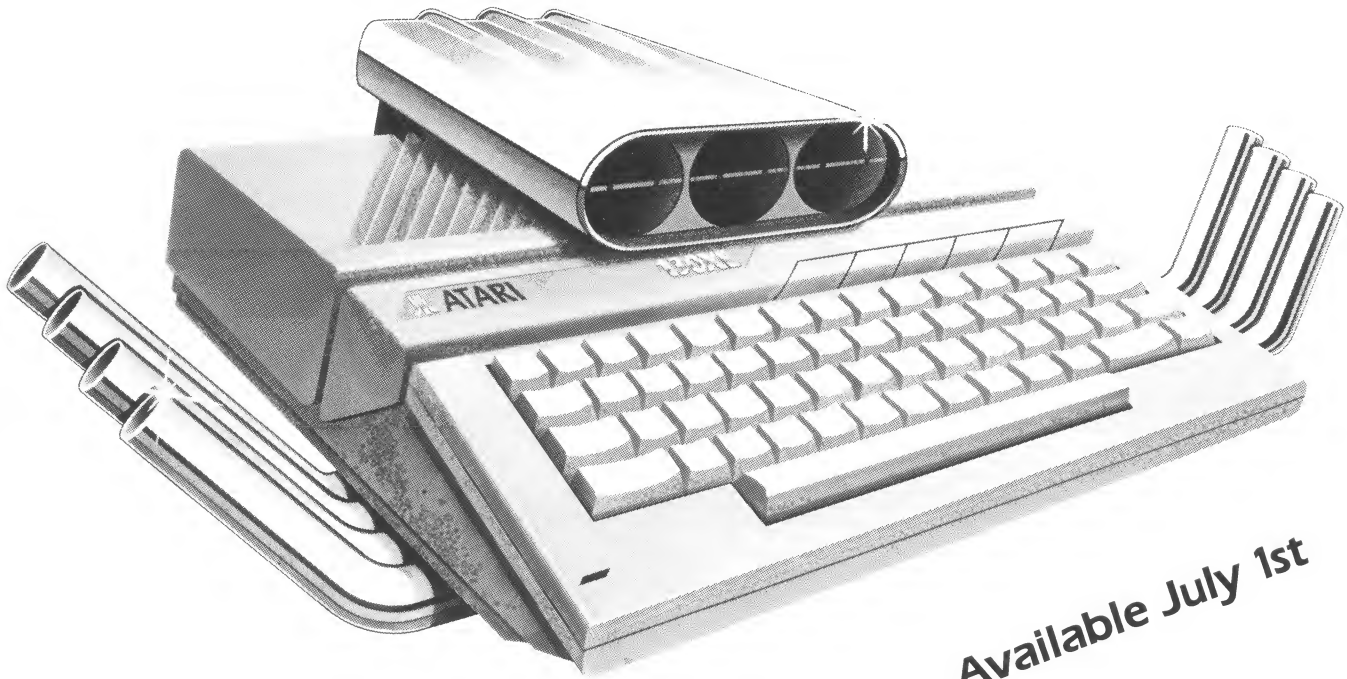


PRECISION SOFTWARE
A DIVISION OF O.S.S., INC.

1173D S. Saratoga/Sunnyvale Road
San Jose, CA 95129 • (408) 446-3099

ATARI and APPLE II are trademarks of Atari, Inc. and Apple Computer, Inc., respectively. MAC/65, BUG/65, C/65, BASIC XL, OS/A+, and OSS PRECISION SOFTWARE™ are trademarks of O.S.S. Inc. ACTION! is a trademark of Action Computer Products.

BASIC XE™ Gives Your Atari 130XE™ All The Performance It Should Have Had In The First Place



Available July 1st

In the home computer races, the Atari 130XE stands out as a price leader. But using underpowered Atari BASIC™ on this otherwise fine machine is like racing in the Indy 500 with half your cylinders missing. So don't get left at the starting line with only half an "engine." Change to the performance leader **now!** Buy BASIC XE from OSS, the **only** programming language designed especially for the Atari 130XE.

Just look at what you get for one low sticker price:

BEST MILEAGE: With over 60,000 **more** bytes for your programs, BASIC XE lets you use all the memory you paid for.*

MORE HORSEPOWER: Run Atari BASIC programs 2 to 6 times faster.* Even with its incredible power, BASIC XE is compatible with Atari BASIC.

BETTER HANDLING: With auto line numbering, renumbering, program cross referencing, English error messages, and more.

CLASSIC DESIGN: Show off the sleek **structured** style of your own programs when you use BASIC XE statements like PROCEDURE, IF...ELSE, and WHILE...ENDWHILE.

FREE ACCESSORIES: Get over \$100 worth of Atari BASIC options **FREE** when you buy BASIC XE: complete Player/Missile Graphics support, string arrays, DOS access, SORT commands, readable listings...over 50 extras at no additional charge.

- If you're ready to step up to real performance...YOU need BASIC XE now!
- If you haven't written your first BASIC program...YOU need BASIC XE now!
- If you're already a real pro in BASIC...YOU need BASIC XE now!
- **BASIC XE may well be the best buy any Atari owner ever made.**

*Want to know more? Call or write for free brochure or ask your local dealer.
Atari 130XE™ and Atari BASIC™ are U.S. registered trademarks of Atari Corporation.



Optimized Systems Software, Inc.
1221B Kentwood Avenue, San Jose, California 95129 (408) 446-3099

PERSONAL PASCAL

Simple But Professional

Personal Pascal is more than just another compiler, it's a complete software development system! A system which is simple enough for the beginner, yet powerful enough for the professional.

Simple because the most important features of the system are a mouse-click away, thanks to the exclusive Pascal Manager. Professional because the manager gives full access to a program editor, powerful compiler, fast linker, and both standard and GEM-oriented libraries.

Simple keystrokes—often with easy-to-find function keys—command such asked-for features as automatic file backup. Enjoy the convenience of professional full-screen editing with block write, file insert, optional automatic indentation, and more.

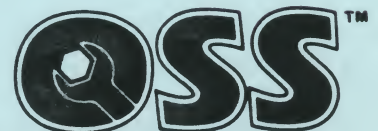
There's more: A simple editor keystroke both saves and compiles your program! Made a mistake? Pop back into the editor and see the error location highlighted, complete with readable error message. Or choose professional options such as modular compilation, debug trace back, and more—all with a few clicks of the mouse. Or use options within a program for even finer control.

Write simple Pascal programs using all the standard keywords and built-in functions. (Even the manual uses the same terminology as one of the popular tutorial books.) Or take advantage of the professional capabilities of advanced string handling, extended I/O, bit-wise operators, external assembly language or C routines . . . the list goes on. Are words such as BIOS, XBIOS, and GEMDOS important? Naturally, they have special meaning to Personal Pascal, also.

Link most programs—even those written with GEM in mind—with a simple mouse-click or two. Or maintain a professional level of control by specifying multiple object files or libraries.

Help with GEM? Of course! A very simple but very effective built-in library provides the best access ever to the most useful parts of GEM. Write a first GEM program within hours (or even minutes) of first using Personal Pascal.

Still there's more! OSS begins its sixth year of providing Atari users with the best products and the best support. Over 275 pages of Personal Pascal reference manual give a first line of support. And OSS stands ready to help registered owners with telephone, bulletin board, and CompuServe.



Optimized Systems Software, Inc.

1221B Kentwood Avenue

• San Jose, California 95129

• (408) 446-3099

PERSONAL PASCAL

Data Sheet

Pascal Manager

- Menu and mouse driven selections
 - Compiler options
 - Linker options
- Call up editor, compiler, linker

Program Editor

- Simple, effective status line
 - Insert and replace modes
- On-line keystroke summary via HELP key
- Full screen, 24 lines by 80 characters
- Logical, simple keystrokes
 - Logical cursor keys (arrows, etc.,) and WordStar-like control keys
 - Move by character, line, or page
 - Optional automatic file backup
 - Optional automatic indentation
- Virtually unlimited copy buffer, copy block or move block
 - Read files into copy buffer
 - Write copy buffer to a file
- Single keystroke to save, exit, and compile current program

Personal Pascal Compiler

- In general compliance with standard Pascal, but with several powerful and useful extensions
- Fast! Single pass compiler
 - May be used with RAM-disk or hard disk
- Flexible, usable arithmetic
 - Six byte (12-digit) floating point
 - Standard 16-bit integer
 - Extended 32-bit long integer
- All standard types, records, sets, user-defined types, subranges, etc.
- All standard operators and functions plus
 - Bit-wise operators (machine level control)
 - Left and right shift functions
- Extended statements for default CASE, simplified LOOP, etc.
- Powerful extended STRING data type
 - String insert, delete, copy, substring search
 - Multiple string concatenation
- Standard I/O, including WRITELN, GET, PUT, file pointers, etc., plus
 - Dynamic file name assignment
 - Random access files
 - File erase and rename
 - Controllable error handling

- Several kinds of external routines supported
 - Modular Pascal compilation
 - C and assembly routines callable
- BIOS, XBIOS, GEMDOS directly callable
 - Standard pointers, NEW, DISPOSE, MARK, etc.
 - Special BASEPAGE pointer type
 - Pointer range checking may be disabled
- Miscellaneous helpful routines
 - CHAIN to another program
 - Check for a keypress
 - Check memory available
 - Get command line arguments
 - Find size of a data object
- Much, much more!

Pascal Linker

- Links all standard GEMDOS (TOS) files
 - Object code from compiler
 - Library files
 - Result is proper relocatable object code
- Fast! Uses memory intelligently
- Pascal manager provides library linkage for TOS or GEM applications

GEM Library

- Complete, self-contained library
 - No need to buy expensive developer's package
 - No license fees
- Access to most useful GEM routines
 - Alert boxes
 - Dialog boxes
 - Menu bar
 - Window management
 - Window text and graphics
 - Event and message management
 - Mouse control

Reference Manual

- Actually several manuals in one
 - "How to use" manuals for editor, compiler, linker, manager
 - Full GEM/Pascal library reference manual
 - Full Personal Pascal language reference manual
- Over 275 pages, with more info on disk

Price

- Disk and Documentation **\$74.95**

ACTION!™

DESCRIPTION:

ACTION! is a high-level, structured programming language environment for the Atari Home Computer.

REQUIRES:

ACTION! requires a working knowledge of BASIC (or an equivalent language) and an Atari Home Computer with at least 16K of memory.

OPERATING SYSTEM REQUIREMENTS:

ACTION! will work with any DOS that supports the Atari Home Computer. Or use it with cassette alone.

PRODUCT MEDIUM:

ACTION! is available in the specially designed, memory-saving OSS SuperCartridge.

DOCUMENTATION:

ACTION! comes complete with a perfect-bound, 205 page reference manual.

SOFTWARE SUPPORT:

Available separately is the ACTION! TOOLKIT (*see reverse side*) and an ACTION! RunTime package. ACTION! RunTime packages are available for both individual and commercial use.

The ACTION! System is Made Up of Four Different Components:

- The MONITOR
- The EDITOR
- The COMPILER
- The LIBRARY

The MONITOR Functions:

- EDIT, COMPILE, and RUN programs or WRITE your compiled program to disk or cassette
- EXECUTE commands immediately
- Select compiler and editor OPTIONS
- DISPLAY memory, Exit to DOS, and more

The EDITOR Features:

- 240 characters per line, two windows, and fully scrolling display
- READ and WRITE text files to/from disk, cassette, printer, etc.
- UP and DOWN by lines or pages, LEFT to RIGHT by characters or to end of line
- MOVE and COPY text blocks, FIND and SUBSTITUTE text

The COMPILER Features:

- Compilation from either memory, disk or cassette
- Ability to INCLUDE library files
- Compile time substitution via DEFINE

ACTION!'s Built-in Library Includes:

- I/O routines: PRINT, PUT, INPUT, GET, OPEN, CLOSE, X10, and more
- Graphics routines: GRAPHICS, SETCOLOR, PLOT, DRAWTO, FILL, POSITION, SOUND, PADDLE, STICK, and more
- String handling routines: String COPY, String COMPARE, String ASSIGN
- Miscellaneous routines: RANDOM, BREAK, ERROR, and many more

The ACTION! Language:

Data types supported:

- 8 bit BYTEs (or CHARacters)
- 16 bit INTegers and unsigned CARDinals
- ARRAYs of and POINTERs to BYTE, INT, and CARD
- User-defined TYPEs (records) and POINTERs thereto

Language Constructs:

FUNCTions (which can RETURN values) and PROCedures

- IF...THEN...ELSEIF...ELSE...ENDIF
- DO...UNTIL, WHILE, and FOR (with STEP), all with optional exits

ACTION! TOOLKIT

DESCRIPTION:

The ACTION! TOOLKIT is a programming aid designed to assist the ACTION! programmer in maximizing the capabilities of the ACTION! language.

REQUIRES:

The ACTION! TOOLKIT requires an ACTION! SuperCartridge and any DOS that supports the Atari Home Computer. Memory Requirement—48K.

PACKAGING:

The ACTION! TOOLKIT is available on diskette only and comes complete with instruction manual.

INCLUDES:

- Player/Missile graphics routines
- A floating point package
- A sorting routine for sorting BYTES, CARDS, INTs, and string data
- Routines that allow dynamic runtime memory manipulation
- A routine that will return the absolute value of an INTEger
- Routines which implement advanced I/O operations
- A music demo which creates a playable organ
- Demo programs
- Games
- And more!

TESTIMONIES

"For those who have found BASIC to be too slow or assembler too difficult, ACTION! is the logical alternative. ACTION! programs can increase speed from 50 to 200 times that of BASIC."

Jerry White, *ANTIC*, February 1984

"Assembly programmers will appreciate the extraordinary efficiency of the ACTION! compiler. The code...uses no special stacks of indirect pointers to control the flow of execution, just pure in-line machine code with an occasional JSR into a cartridge library routine. This is "native mode" compilation at its best: Simple, clean, and very, very swift. The output of a typical C or PASCAL compiler looks like spaghetti by comparison."

Brian Moriarity, *Analog Computing*, Issue 16

"Programs written in ACTION! very much resemble those written in PASCAL, both in elegance and efficiency....These features mean that well-written programs in ACTION! will be easy to read and maintain....ACTION! has the best ratio of sophistication to performance of any language I have seen on *any* microcomputer,"

William C. Covey III, *Educational Technology*

"ACTION! is a structured, compiled language similar to PASCAL and C. It features an excellent screen editor with two windows, a monitor program and a procedure and function library....The ACTION! editor is so well designed that, with minor modifications, we would consider it a product in its own right. The editor alone is worth the price of the language. ACTION! is a well-integrated package. Intermediate and experienced programmers will find it easy to write ACTION! programs quickly."

Harold E. Striepe, *InfoWorld's Essential Guide to Atari Computers*, 1984



Optimized Systems Software, Inc.

1221B Kentwood Avenue, San Jose, California 95129 (408) 446-3099

BASIC XL™

DESCRIPTION:

BASIC XL is a compatible extension to Atari BASIC that adds dozens of new commands and functions. BASIC XL is LOAD compatible with Atari BASIC and supports *all* Atari BASIC commands.

REQUIRES:

BASIC XL requires no previous programming experience and will work with any Atari Home Computer with at least 16K of memory.

OPERATING SYSTEM REQUIREMENTS:

BASIC XL will work with any DOS that supports the Atari Home Computer. Or use BASIC XL with cassette alone.

PRODUCT MEDIUM:

BASIC XL comes to you in the specially designed, memory-saving OSS SuperCartridge.

DOCUMENTATION:

BASIC XL comes complete with a perfect-bound, fully-documented reference manual.

SOFTWARE SUPPORT:

Available separately for BASIC XL is the BASIC XL TOOLKIT (*see reverse side*). Available for beginners, is our highly-accredited tutorial "30 Steps to Understanding BASIC XL."

SUMMARY OF NEW FEATURES:

- Dazzling Player/Missile Graphics
- Improved String Handling
- Complete Program Control
- Increased Program Development
- Versatile Input/Output
- Special System Options

Player/Missile Graphics Support:

- Commands include: PMCOLOR, PMMOVE, PMGRAPHICS, PMWIDTH, and MISSILE
- Ability to efficiently MOVE a large player vertically with a single command
- Detect collisions with the specially designed BUMP command

String Handling Capabilities:

- Support of String Arrays
- String functions include: LEFT\$, RIGHT\$, and MID\$
- A FIND function, to find a match of one string within another

Program Control:

- Better program control with statements like IF...ELSE...ENDIF, and WHILE...ENDWHILE
- Move Bytes around at assembly language speed with BASIC XL's MOVE command

Special Program Development Commands:

- Use the NUM and RENUM commands to enable BASIC XL's automatic line numbering
- Run lengthy programs 2 to 4 times faster with BASIC XL's specially designed FAST mode
- Make debugging easier with the TRACE/TRACEOFF commands and English error messages (no more looking up what "ERROR 133" is)
- Get a cross-reference of program variables in use with the LVAR command

Comprehensive Input/Output:

- Handles DOS commands like DIR, PROTECT, ERASE, and RENAME right from BASIC XL
- Use RGET and RPUT commands for record I/O
- Support a PRINT USING command for sophisticated output formatting
- Manipulate data blocks from a device or file with BPUT and BGET commands
- Utilize the TAB function to make column printing easier

Special System Options:

- Catch run-time errors and lines using the ERR function
- Look and change two bytes instead of one with DPEEK and DPOKE
- Use hexadecimal numbers at any time or get the hexadecimal equivalent of decimal numbers with the HEX\$ function

BASIC XL TOOLKIT

DESCRIPTION:

The BASIC XL TOOLKIT is an excellent programming aid, designed to work in conjunction with the OSS BASIC XL SuperCartridge.

REQUIRES:

The BASIC XL TOOLKIT requires a BASIC XL SuperCartridge and any DOS that supports the Atari Home Computer. Memory Requirement—48K.

PACKAGING:

The BASIC XL TOOLKIT is available on diskette only and comes complete with a comprehensive instruction manual.

HERE'S WHAT'S INCLUDED:

- The BASIC XL TOOLKIT comes with a copy of the BASIC XL RunTime package. This means that you can run BASIC XL programs without having the BASIC XL SuperCartridge present in your computer.
- The BASIC XL TOOLKIT is filled with sample BASIC XL programs that are designed to show off the speed and versatility of the BASIC XL language. Programs such as these will aid the serious programmer in writing quality programs with professional results.
 - PHONE: a simple data base manager using indexed files
 - DISKIO: direct access to disk sectors
 - LEM: a lunar lander using Player/Missile Graphics
 - CIRCLES: a simple circle drawing routine
 - SNAILS: a simple game demonstrating some of BASIC XL's Player/Missile Graphics
 - Plus Much More!

NEW EXTENDED COMMANDS:

- The BASIC XL TOOLKIT adds these new Extended commands to the already packed BASIC XL language.
 - PROCEDURE: defines a block of PROCEDURE code
 - EXIT: declares the end of a PROCEDURE
 - CALL: used to CALL a PROCEDURE
 - LOCAL: defines LOCAL variables within a PROCEDURE
 - SORTUP: SORTS string arrays in ascending order
 - SORTDOWN: SORTS string arrays in descending order

TESTIMONIES

"If you have any designs at all on writing your own programs, run, do not walk, to your nearest dealer and buy a copy of BASIC XL. It is flat out one of the best additions I have ever found for my computer."

Rob Peck, *MC2 Michigan Computer Consortium*, February 1984

"BASIC XL is the fastest and most powerful version of BASIC available for Atari Computers. If you program in BASIC, or if you would like your BASIC programs to run faster, I highly recommend this language....A truly outstanding computer deserves a truly outstanding BASIC. This is the language that should be built into Atari Computers."

Jerry White, *ANTIC*, March 1984

"BASIC XL is a fast and powerful extension of Atari BASIC, totally compatible with virtually all software. Its many features make programming easy, especially games that require player/missile graphics. For people writing business software or translating existing programs from other computers, the new string arrays and other string handling features make the task manageable. BASIC XL is a truly professional language that should become standard in all future Atari computers. Overall Rating A."

The Addison-Wesley Book of Atari Software 1984

"I have to say you have accomplished what I hoped for and far exceeded my hopes. I had hoped for a descent BASIC but you have produced what appears to be a GREAT BASIC."

Paul E. Phelps, Chaplain (Maj), USA



Optimized Systems Software, Inc.

1221B Kentwood Avenue, San Jose, California 95129 (408) 446-3099

THE WRITER'S TOOL™

DESCRIPTION:

THE WRITER'S TOOL is a complete word processing package designed for *anyone* who wants to write. Its ease of use and flexibility are excellent supporting features. But the most significant feature of THE WRITER'S TOOL is its capability to produce "professional looking" documents.

REQUIRES:

THE WRITER'S TOOL requires no previous word processing experience and a Atari Home Computer with at least 48K of memory. An Atari compatible disk drive is also required to operate THE WRITER'S TOOL.

OPERATING SYSTEM REQUIREMENTS:

THE WRITER'S TOOL is so versatile, that it will work with any Atari compatible DOS in any density.

PRODUCT MEDIUM:

THE WRITER'S TOOL comes to you on an OSS SuperCartridge and a two-sided disk.

DOCUMENTATION:

THE WRITER'S TOOL comes complete with an excellent tutorial, reference manual, and reference card.

SUMMARY OF KEY FEATURES:

- Supports all popular printers
- Includes an integrated spelling checker
- Contains a mail merge subsystem
- Complete and easy-to-follow tutorial

PRINTER SUPPORT:

- THE WRITER'S TOOL comes with a pre-written set of printer support files for most popular printers, including printers by:
 - Atari Corporation
 - Epson, Inc., and compatibles (Gemini)
 - C. Itoh, Ltd., and compatibles
 - Okidata
 - And more!

PRINTER SUPPORT (Continued):

- THE WRITER'S TOOL also includes a special file that will allow you to create your own custom printing file for virtually any printer. With THE WRITER'S TOOL you will be able to utilize the special features of your printer.

FEATURES:

- Typeover and Insert modes
- Delete and Restore
- Search and Replace
- Headers and Footers
- Page numbering
- Centering
- Pica, Elite, Condensed, and Proportional font support
- Micro spacing
- Subscripts and Superscripts
- Split justify
- Soft hyphen and hard space
- Optional word wrap
- Outdenting
- Boldface and underline
- Straight or ragged right justification
- Plus more!

SPELLING CHECKER:

- THE WRITER'S TOOL spelling checker is designed to quickly scan your document and flag and correct any words that are misspelled.
- Includes a 20,000+ word dictionary.
- Allows you to create any number of custom dictionaries for those special documents that may contain technical or unusual words.

MAIL MERGE SUBSYSTEM:

- Use the built-in mail merge subsystem to give your letters that "personal" touch.
- Include files from data bases, such as SynFile+, to save yourself valuable time.

TESTIMONIES

"THE WRITER'S TOOL is probably the best compromise between total word control and ease of use that I've seen in a word processor. THE WRITER'S TOOL is loaded with features; in fact, it has most of the features found in the "professional" word processors, plus a few tricks of its own. But it's an easy package to learn and even easier to use. Is there anything I don't like about THE WRITER'S TOOL? Actually, no...."

Bob Curtin, *ANALOG Computing*, March, 1985

"THE WRITER'S TOOL goes beyond most home computer word processors in its amazing breadth of features while still remaining easy to learn and use. At last I've found a piece of software which will support *anything* my printer is capable of, and yet is still easy to use. Bottom line—If you have a printer that's capable of some tricks, and your word processing needs go beyond simple letter writing, I strongly urge you to visit your software dealer and give THE WRITER'S TOOL a look. Do this even if you already own a word processor, this is one hot product!"

Bob Peck, *The Michigan Computer Consortium Magazine*, December, 1984

"Up to this point, I have been impressed with the ease of using this program. A statement on the manual-cover claims that THE WRITER'S TOOL is the most natural and complete word processor available for the Atari. They may be right. The author is to be congratulated for writing one of the best computer-related manuals to be had. This is a full-function word processor, that is incredibly easy and natural to use. This is the best word processor that I have used on the Atari."

Jeff Golden, *Dallas ACE Newsletter*, January, 1985



Optimized Systems Software, Inc.

1221B Kentwood Avenue, San Jose, California 95129 (408) 446-3099

Personal Prolog...

The Language for Intelligent Programming

Discover how easy the world of Logic Programming can be with **Personal Prolog**, a sensational new "Fifth Generation" language for your Macintosh. **Personal Prolog** is more than just another programming language: it's a complete multi-window programming environment-- with integrated editor, interpreter, debugging aids, and all the built-in support you need to educate yourself in the world of Artificial Intelligence.

Because **Personal Prolog** is fully integrated with the Macintosh desk-top environment, it's ease of use is greatly enhanced by allowing you to utilize pull-down menus. The Editor conforms to the standard Macintosh interface and allows you to edit up to seven windows at one time. **Personal Prolog** also extends the standard language by providing an extensive interface to the Macintosh ToolBox, allowing you to add graphics and mouse control to your Artificial Intelligence applications.

Use **Personal Prolog** to draw conclusions, interpret human language, create expert systems, examine mathematical logic, and much more! Buy **Personal Prolog** today and discover why Prolog is fast becoming a worldwide favorite... the Language for Intelligent Programming.

Only ... \$74.95

Requires Macintosh (512K)

The following is an extraction from a published review of **Personal Prolog**:

"**Personal Prolog** is a nicely designed, well-implemented version of the increasingly popular AI programming language. **Personal Prolog** is aimed at those who want to learn PROLOG and specifically not at developers. As such, it offers one of the best price/performance trade-offs I've seen in any dialect of PROLOG. As a Macintosh product, **Personal Prolog** draws very high marks. It adheres closely to Apple's standard Mac user interface, which will make it a very comfortable environment for first-time PROLOG users. The pull-down menus, dialog boxes, windowing, scroll bars, and other familiar Mac features are used as expected. In addition, **Personal Prolog** gives direct access to all the Mac's graphics and QuickDraw routines. Making graphics calls in **Personal Prolog** is as easy as making calls directly to the routines but they are preceded by an @ sign.

Personal Prolog's menus are clean, easy to understand, and for the most part operate in one step (that is, they don't invoke further layers of menus but instead carry out functions). The editor used in **Personal Prolog** is the Macintosh text editor, which permits selection, movement, alteration, and deletion of text using the mouse and keyboard in combinations comfortable to the Mac user.

On the whole, **Personal Prolog** is a very good implementation of the language for the Macintosh. It is consistent, easy to use, predictable, and well thought out. The bottom line is if you have a Macintosh (or soon an Atari ST) and you're curious about PROLOG, you don't have to wait for Borland to get around to putting Turbo Prolog on your favorite system. **Personal Prolog** is sufficiently powerful for a solid learning experience, offers a strong Mac-like interface, and provides excellent performance for the price. I highly recommend it."

Dan Shafer, "Personal PROLOG," **COMPUTER LANGUAGE**, Volume 3, Number 9 (September 1986)



Optimized Systems Software, Inc.

1221B Kentwood Avenue, San Jose, California 95129 (408) 446-3099



Personal Prolog™

The Personal Prolog System

- Complete Macintosh Interface compatibility
- Utilize up to seven Editing windows
- Unlike most versions of Prolog, Personal Prolog includes a separate History Window
- "Ask" Window allows you full query control by providing you with multiple spy options

The Personal Prolog Language

Parts of Speech

- Constants: Numbers, Strings, and Atoms
- Variables: Standard naming conventions, plus decompiled programs retain your assigned names
- Symbols: Flexible usage. The Personal Prolog standards are: "if", "and", and ":", but you may use various equivalents, including ":-", ":", and "!", respectively

Speech Syntax

- The Data Base: Unlike most Prologs, the Data Base itself is an accessible list thus sidestepping the need for 'functor' and providing a high degree of flexibility
- Lists: No restrictions on the elements of a list; mixed lists are easy to process
- Predicates and their Clauses: Complete predicates or individual clauses may be inserted into or deleted from the Data Base using standard built-ins

Personal Prolog Built-ins

Data Base Management Built-ins

- Search the Data Base for a specific predicate or clause in a predicate
- Control the matching of predicates
- Insert or Remove both rule and fact clauses from a predicate
- Provides a method for reexecuting a call list that would otherwise fail on backtracking

Term Management Built-ins

- The ability to count the number of elements in a list
- Determine the type (class) of a term (i.e., number, atom)
- Convert a non-list term to a list or vice versa

Term Comparison and Math Built-ins

- Determine the absolute value of a number
- Compare the values of two constants
- Determine the remainder left over when a number is divided by another
- Determine the product or quotient in multiplicative operations
- Perform the additive operations (addition, subtraction)

Input and Output Built-ins

- Read clauses from a disk file and compile them into the Data Base
- Read or Write a character to a channel opened for input or output
- Print the hexadecimal equivalent of a number to a channel opened for output
- Open a channel to a disk file for input from or output to that file
- Decompile a predicate from the Data Base and prints it to a channel

Personal Prolog Graphics

- Special Personal Prolog graphics features
- Access to most useful Macintosh Toolbox routines
- Access to both Text and Graphics Windows
- Utilize different variations of the Graphics Pen
- Drawing simple shapes (circles, rectangles, etc.) has never been easier
- Construct such complex shapes as Polygons, Regions, and Pictures
- Complete Mouse Control

Complete With Documentation

- Get started immediately with a special "Hands On" section designed for people who have never used Prolog
- Also includes a very comprehensive reference manual, complete and fully indexed
- Special Graphics section for including graphics within your Personal Prolog programs

Personal Prolog™...

The Language for Intelligent Programming

Discover the world of logic programming with **Personal Prolog**, a sensational new "Fifth Generation" language for your Atari ST, from the people at **OSS**. **Personal Prolog** is more than just another programming language: it's a complete multi-window programming environment—with integrated editor, interpreter, debugging aids, and all the built-in support you need to explore the world of artificial intelligence.

Personal Prolog is such a flexible language that it will accept most forms of Marseilles and Edinburgh syntax. Because it is fully integrated with the GEM desk-top environment, **Personal Prolog** is easy-to-use, with pull-down menus and full mouse control.

Personal Prolog also extends the standard language by providing you with an extensive interface allowing you to add graphics and mouse control to your Artificial Intelligence applications. The **Interpreter** supports incremental compilation and is significantly faster than other Prolog systems designed for the IBM PC. Editing a program has never been easier, as the **Editor** is both easy-to-use and extremely efficient.

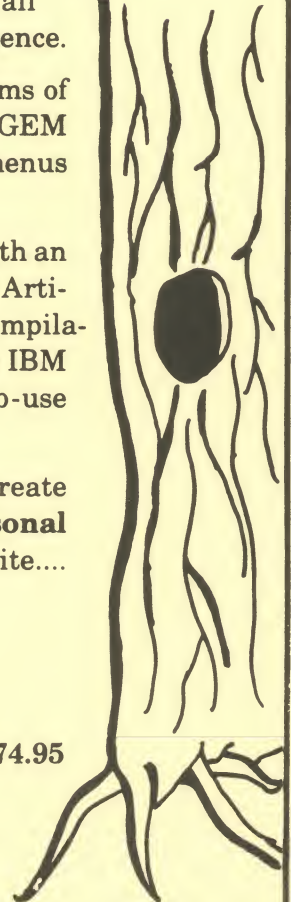
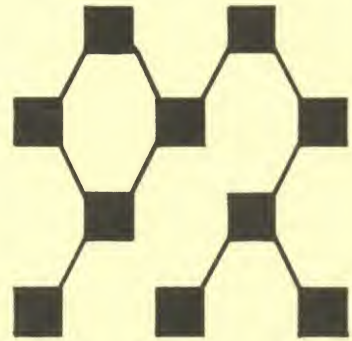
Use **Personal Prolog** to draw conclusions, interpret human language, create expert systems, examine mathematical logic, and much more! Buy **Personal Prolog** today and learn why Prolog is fast becoming a worldwide favorite.... the language for intelligent programming.

Special Introductory Price \$74.95



Optimized Systems Software, Inc.

1221B Kentwood Ave., San Jose, CA 95129, (408) 446-3099, Telex: 5106016124



BASIC XE FROM O.S.S.

Ultimate BASIC for 8-bit Atari

by CHRISTOPHER CHABRIS

When Atari shipped the 130XE computer in April 1985, Optimized Systems Software (OSS) seized the opportunity to create the first programming language designed for the expanded 128K memory of the XE. The result is cartridge/disk **BASIC XE**, the fastest and most powerful BASIC available for 8-bit Atari XL and XE computers. NOTE: BASIC XE will *not* run on Atari 400 or 800 models. Users of these earlier computers still have BASIC XL available from OSS.

Optimized Systems Software was the first independent company to write software for Atari personal computers. Bill Wilkinson and his programmers created the disk file management system of DOS 1.0, 2.0 and 2.5, as well as Atari BASIC and the Atari Assembler Editor cartridge. But they didn't stop with software just for Atari to bundle with its computers. OSS went on to publish the best line of Atari programming languages and tools including MAC/65, ACTION! and BASIC XL.

BASIC A+ was the first OSS enhanced version of Atari BASIC. Unlike the Microsoft BASIC released by Atari, BASIC A+ was compatible with source code that ran under Atari BASIC, except for a few minor exceptions. BASIC A+ was available only on disk, but the subsequent, improved BASIC XL came in super-cartridge form, using the technique of bank selection to cram 16K of ROM code into 8K of address space and save 8K for programs.

Now comes BASIC XE, one of the largest programming languages for the 8-bit Atari computers. Its 27K of code is divided between a 16K super-cartridge and 11K of extensions that can be loaded from disk when the cartridge is booted. If these optional extensions are not present on the boot disk, BASIC XE still functions—but without some of its extra features. With or without the extensions, the language occupies 8K of address space.

CUSTOMIZED STATEMENTS

BASIC XE is rich in commands and functions, with 140 built-in keywords. I say "built-in" because the language makes it possible for programmers to create named procedures that accept parameters—in effect, adding statements to the language. This enhancement completes the evolution of Atari BASIC into a tool for modern programming. The IF:ELSE:ENDIF and WHILE:ENDWHILE constructs (introduced in BASIC A+) allow for Pascal-like structured programming style, and the new PROCEDURE:LOCAL:EXIT trio coupled with CALL give the programmer modular design and recursion capabilities in an interpreted language.

It is now possible to create libraries of often used PROCEDURES quite simply. Before BASIC XE, the programmer had to precisely define which variables contained values to be passed to a subroutine, which variables would change during its execution, and which variables would contain values returned from the subroutine. A chore, and difficult to debug! Here's a sample PROCEDURE definition and execution in BASIC XE:

```
30000 REM CALCULATE F(X) WHERE F IS A
      POLYNOMIAL WITH COEFFICIENTS IN P()
30010 PROCEDURE "EVALPOLY" USING Degree
      e, !P(), X:LOCAL E, V
30020   FOR E=Degree TO 0 STEP -1
30030     V=V+(X^E)*P(E)
30040   NEXT E
30050 EXIT V
```

These lines would fall near the end of the program, usually beyond the END statement. They define a new procedure named EvalPoly (Evaluate Polynomial) which receives three arguments: Degree, the degree of the function; P(), an array containing the coefficients in ascending order; and X, the value at which to evaluate P. Note

continued on next page

that non-scalar parameters are preceded by an exclamation point.

LOCAL creates the following variables as temporary scalars. Other variables with the same names are not affected, and the local variables disappear when an EXIT is encountered. The EXIT statement causes the procedure to return V (making the procedure really a function!). EvalPoly could be called as follows:

```
CALL "EvalPoly" USING 4,IF(),2.5 TO Y
```

This statement would evaluate the fourth degree polynomial represented in array F at the value 2.5 and store the result in variable Y. The beauty of this parameter passing and returning scheme is that it is totally independent of line numbers and variable names. It is portable like Pascal procedures or C functions.

I foresee the availability of specialized libraries to add capabilities to BASIC XE. For example, you could collect procedures to do matrix algebra or create various graphics objects. Since LOCAL can only create scalar floating point variables, BASIC XE is particularly suited to mathematical applications.

SPEED TO SPARE

This brings us to another new feature of BASIC XE—its fast mathematics routines which replace those built into the XL/XE Operating System, when the extensions are loaded. According to OSS, these routines are more accurate and twice as fast as those in the FastChip from Newell Industries, which are supposed to be 30% faster than Atari's built-in routines.

BASIC XE also features the semi-compiled mode of operation introduced in BASIC XL. When the FAST command is encountered at the beginning of a RUNning program, all line number references are converted into absolute addresses, eliminating many time-consuming searches through the program. OSS claims that these two enhancements make BASIC XE run two to six times faster than Atari BASIC.

To test BASIC XE's speed, I used the off-the-shelf Atari BASIC software B/Graph, from Batteries Included. Among B/Graph's statistical capabilities is a program to do regression analyses. I felt this to be a good test because it uses extensive iterations and floating point calculations—BASIC XE's strengths. Using the sample datafile PLANET, I followed the instructions on pages 113—114 of the B/Graph Manual and obtained the following completion times for different versions of BASIC:

Atari BASIC, rev. C	30 seconds
BASIC XL, v1.03	12 seconds
BASIC XE, v4.10	7 seconds

The first was tested under Atari DOS 2.05, the others under OSS DOS XL 2.30p. BASIC XE without the extensions loaded performed similarly to BASIC XL, and using EXTENDED mode made no difference in any case.

So BASIC XE does offer significant improvements in speed when running Atari BASIC programs. Exercising its features from start to finish in the program development

process should yield shorter, more elegant and readable, and still faster code than possible with any other BASIC for the Atari. And it will take less time to write the program because BASIC XE's powerful commands take the place of many machine-language subroutines. All this is possible on either the 1200XL, 600XL with 64K, or 800XL computers. But with the 130XE, you can do even more.

130XE SPECIALS

The new EXTEND command instructs BASIC XE to utilize the extra 64K memory available in the 130XE's secondary bank. Your program itself is relocated into this space. Main memory is reserved for variables, the stack, and other related items. (Page Six is always available for user purposes.) In EXTENDED mode, depending on your Disk Operating System, there will be approximately 63K available for program and 32K available for data.

In this way, BASIC XE lets you take full advantage of your 130XE's memory without worrying about the nitty-gritty details of memory management. However, if you want to use your extra 64K in another way besides a RAM-disk, BASIC XE allows you to specify an optional bank number in the POKE, DPOKE, PEEK, DPEEK, MOVE, BGET and BPUT commands. When referring to an address in the range \$4000-\$7FFF, the secondary bank's "access window", banks 0-3 are within that bank and the default bank 4 indicates main memory. This makes memory management convenient compared to twiddling the PIA PortB bits.

BASIC XE's other advanced features are, as they say, too numerous to even list here completely. They include, among others, commands to sort arrays into ascending or descending order, Microsoft BASIC string handling, OSS's classic file manipulation and Player/Missile graphics commands, bit-manipulation operators, hexadecimal numeric support, and program development aids like disk directory, renumbering, and cross-referencing.

What should be added to BASIC XE? Well, using the 5K remaining in my wished-for 32K cartridge, OSS could add the REPEAT:UNTIL construct, integer variables, dynamic memory allocation, and advanced sound control—to name just a few possibilities. And of course, some sort of run-time library or compiler would be nice.

BASIC XE is an excellent product. It's the language that should have been built into the 130XE. OSS is a company that has always supplied the highest quality systems software for all Atari computers, but they have outdone themselves with this one. One can only wonder how they will top BASIC XE. I hear they are developing for the ST machines.

BASIC XE
Optimized Systems Software
1221B Kentwood Avenue
San Jose, CA 95129
(408) 446-3099
For XE/XL Ataris
Cartridge plus disk
\$79.





Optimized Systems Software, Inc.

1221 B Kentwood Avenue, San Jose, CA 95129 (408) 446-3099

To: Registered Personal Pascal Customers

From: Customer Support Group

Please find enclosed disk containing supplemental documentation for Personal Pascal. Included on this disk are many sample programs showing how to use Personal Pascal for various applications.

We hope this will help you take full advantage of the power and flexibility of Personal Pascal.

We realize that the documentation for Personal Pascal does not cover all possible situations. To help rectify this, we have set up our own BBS system for use by our customers. This BBS system has many examples covering a wide variety of subjects. We generally try to give examples on the most common problems. You may also use the BBS to leave messages to the Customer Support Group after hours and on weekends. The number to call for the BBS is (408) 446-3451, 300 or 1200 Baud 24 hours / 7 days a week.

If you have an account with Compuserve, you may send E-MAIL to us at 72477,3703. There are many example programs written in Personal Pascal in the ATARI16 sig; Data Library 3 (DL3). Use the BROWSE command with a keyword of PASCAL or OSS.

Regards,
Optimized Systems Software, Inc.

Michael Curry
Technical Support Representative

OSS Products
are available from this dealer

- C/65
- tiny-c
- OS/A+
- MAC/65
- BASIC A+
- SpeedRead+

Optimized Systems Software, Inc.
10379 Lansdale Ave., Cupertino, CA 95014
(408) 446-3099

SYNTAX SUMMARY
for BASIC A+ and ATARI BASIC

STATEMENTS

*Denotes features found ONLY in BASIC A+
†Denotes statements not in Apple version

*BGET #fn, addr, len
*BPUT #fn, addr, len
BYE
†CLOAD
CLOSE #fn
CLR
COLOR aexp
CONT
*CP
†CSAVE
DATA (ascii data)
DEG
*DEL line [line]
DIM svar (aexp)
DIM mvar (aexp [aexp])
*DIR filename
DOS
*DPOKE addr, aexp
DRAWTO aexp, aexp
*ELSE {see IF}
END
*ENDIF {see IF}
*ENDWHILE
ENTER filename
*ERASE filename
FOR avar=aexp TO aexp
[STEP aexp]
GET #fn, avar
GOSUB line
GOTO line
GRAPHICS aexp
IF aexp THEN (stmts)
IF aexp THEN line
*IF aexp : (stmts)
ELSE : (stmts)
ENDIF
*INPUT "...", var
[var...]
[var...]
INPUT [var...]
[var...]
[LET] svar=sexp
[LET] svar=aexp
[LET] mvar=aexp
LIST [filename]
LIST [filename], line
LOAD filename
LOCATE aexp, aexp, avar
*LOMEM addr
LPRINT [exp [exp...]]
[exp...]
*LVAR filename
†*MISSILE pm, aexp, aexp
*MOVE fromaddr,
toaddr, lenaexp
NEW

NEXT avar
NOTE #fn, avar, avar
ON aexp GOTO line
[line...]
ON aexp GOSUB line
[line...]
OPEN #fn, mode, avar,
filename
PLOT aexp, aexp
†*PMCLR pm
†*PMCOLOR pm, aexp, aexp
†*PMGRAPHICS aexp
†*PMMOVE pm [aexp]
[aexp]
†*PMWIDTH pm, aexp
POINT #fn, avar, avar
POKE addr, aexp
POP
POSITION aexp, aexp
PRINT [var...]
PRINT exp [exp...]
[exp...]
PRINT #fn [exp...]
[exp...]
*PRINT #fn, USING sexp,
exp [exp...]
*PROTECT filename
PUT #fn, aexp
RAD
READ var [var...]
REM (any remark)
*RENAME filenames
RESTORE [line]
RETURN
*RGET #fn, asvar
[asvar...]
*RPUT #fn, exp [exp...]
RUN [filename]
SAVE filename
*SET aexp, aexp
†SETCOLOR aexp, aexp, aexp
†SOUND aexp, aexp
aexp, aexp
STATUS #fn, avar
STEP (see FOR)
STOP
*TAB [var...], avar
THEN (see IF)
TO (see FOR)
*TRACE
*TRACEOFF
TRAP line
*UNPROTECT filename
*WHILE aexp
XIO aexp, #fn, aexp,
aexp, filename
? (same as PRINT)

FUNCTIONS

ABS(aexp)
ADR(svar)
ASC(sexp)
ATN(aexp)
†*BUMP(pm, aexp)
CHR\$(aexp)
CLOG(aexp)
COS(aexp)
*DPEEK(addr)
*ERR(aexp)
EXP(aexp)
*FIND(sexp,
sexp, aexp)

FRE(0)
†*HSTICK(aexp)
INT(aexp)
LEN(aexp)
LOG(aexp)
PADDDLE(aexp)
†*PEN(aexp)
†*PMADR(pm)
PTRIG(aexp)
PEEK(addr)

RND(0)
SGN(aexp)
SIN(aexp)
SQR(aexp)
†STICK(aexp)
†STRIG(aexp)
STR\$(aexp)
*TAB(aexp)
USR(addr
[aexp...])
VAL(sexp)
†*VSTICK(aexp)
*SYS(aexp)

MORE PROGRAMMING AIDS

MEANINGFUL ERROR MESSAGES

Now no need to look up those error numbers. BASIC A+ tells you what the problem is.

TRACE

Use the TRACE command to follow your program flow as each line is executed.

SET and SYS

Allows the BASIC A+ user to change and examine system parameters, such as: disallow breaks, change tab width for print, and more.

INPUT "..."

Output a prompt string and request keyboard input with a single statement.

AND EVEN MORE!!!

There's more. More than we can possibly show here. Things like optional zero-time for loops, optional lower case input, easy program chaining, and overlays. And more.

No other Basic for Atari can match BASIC A+ when it comes to features, compatibility, and ease of use.

P.S. And, of course, BASIC A+ is available to those Apple II programmers who need a business-oriented BASIC. (But caution: BASIC A+ is not compatible with Applesoft.)

BUG/65

Now get the most powerful debugger yet for your Atari or Apple computer. **BUG/65** includes all the traditional debugging operations: display memory, change memory, disassembly, instant assembly and more. But **BUG/65** shows its real power with such features as: a breakpoint capability that allows for "conditional breakpoints" (i.e., breakpoints that only happen if some register has some particular value, etc.), and a single step and trace mode that even displays the status register in a readable fashion. **BUG/65** is a stand-alone program which allows you to read files into memory (optionally with an offset), write files, and read or write single sectors. Still not enough? How about allowing **BUG/65's** output to go only to the printer, so any display you have on the screen will not be affected. And one more unique and powerful feature: **BUG/65** can be loaded ANYWHERE in memory, so as not to disturb your program.

So why not make **BUG/65** part of your software tools, and make debugging (almost) fun for a change!

MAC/65

MAC/65 is a product that is uniquely OSS. It is the most logical upgrade for either the Atari Assembler-Editor Cartridge or our earlier EASMD.

The EDITOR

MAC/65's editor is the same familiar line oriented editor that comes with both the cartridge and EASMD. Excepting that now there are two edit modes. TEXTMODE allows you to enter code like any other editor. EDIT mode though, is the FIRST difference you'll notice about MAC/65. When in EDIT mode the editor does syntax checking! That's right, every source line is checked for proper assembly language syntax when it is entered.

The ASSEMBLER

Again, MAC/65's assembler can handle everything the cartridge and EASMD can. Such as source code in memory or on disk, object code in memory or on disk, etc. Even all the mnemonics are the same. But the resemblance is only skin deep: Ask for an assembly and watch MAC/65 come into its own! MAC/65 can do memory to memory assemblies at the rate of thousands of lines per minute. Even disk to disk assemblies proceed at hundreds of lines per minute...over 25 times faster than the cartridge. The most unique thing about MAC/65 is its macro power. Now you can easily find out how many arguments were passed to a macro, extract the length of a literal string argument, find out if a label has been defined and/or referenced, and much more. Naturally you may use nested macros, powerful conditional assembly directives (IF, ELSE, ENDIF), usable listing controls, included files, and other control and formatting directives. MAC/65 also features a complete and comprehensive array of operators such as: +, -, /, &, <, >, <=, >=, =, <>, !, .AND, .OR, .NOT, and more, including low-byte-of-address and high-byte-of-address in accordance with MOS Technology standard usage.

But That's Not All

Not only do you get the syntax checking editor and the powerful macro assembler, you also get **BUG/65**, a unique and powerful debugger that alone sells for **\$34.95**. See the description of **BUG/65** in this brochure.

OS/A+

What can you do with an operating system that gives YOU control? PLENTY! With **OS/A+** you can add your own device drivers, add your own commands, and access all system features from C/65 or the assembly language level almost as easily as from BASIC A+.

OS/A+ is our name for an operating system that is interfaced to the user via a general purpose, keyboard oriented, command processor. In addition to several powerful intrinsic commands, any system utility program may be invoked simply by entering its name. And, of course, the utility may examine the invoking command line and process parameters, filenames, etc. Even our power BASIC A+ is treated as a utility by **OS/A+**!

Underneath the simplicity and flexibility of the **OS/A+** console processor lies the real power of the OSS operating system. A truly device independent user interface simplifies the application programmers task: there are no special calls for the console, printer, or disk because all device and files are treated alike. A program can get a byte from the keyboard, a disk file, or a serial port with exactly the same OS command format—only the file (device) name need change.

And now available from OSS is **Version 4 OS/A+** for use on double density (and larger) disk drives such as those from PERCOM, MPC, SOFTWARE PUBLISHERS, and even CORVUS (Apple II Corvus only). What does that mean to Apple and Atari owners? Simply stated, now there is a DOS which can support any size disk drive, up to 16 megabytes or more, and which provides TRUE RANDOM ACCESS file support.

SYSTEM UTILITIES

Included with both versions of **OS/A+**: Format and initialize new disks, copy files, and duplicate entire diskettes. And YES, you can duplicate or copy your disks. Perhaps the best news is that a simple, workable BATCH capability is standard.

Unless otherwise noted, all OSS products require 48K and at least one disk drive. We recommend 64K for the Apple version of **OS/A+**.

BASIC A +

BASIC A + will rate an A + from any business programmer or Atari user! Upward compatible with Atari Basic, it adds statements and features the enhance the Atari 800's real power, flexibility, and ease of use: Superior I/O features for business and other applications. Additional file manipulation commands. Significant help in program development and debug. Structured programming aids. And MORE! A partial list of the enhancements of **BASIC A +** includes:

PLAYER/MISSILE GRAPHICS

The PM Commands

A full set of special statements and functions allows the **BASIC A +** user to exercise almost complete control over the Atari's incredible player/missile hardware.

PMADR and MOVE

With the ability to obtain the memory address of any player or missile combined with the ability to move any block of bytes from and to anywhere in memory, lighting fast changes are possible with **BASIC A +**. And use BPUT and BGET with PMADR for fast P/M loads from disk.

ENHANCED INPUT/OUTPUT

PRINT USING

Easy-to-use, incredibly flexible, uniquely sophisticated. For business use or just for producing readable, sensible output.

PROTECT, UNPROTECT, RENAME, ERASE, DIR

Use these commands with any filename. Or use wildcard searches to affect all or some of the files on a disk.

RGET/RPUT

Provides fixed field I/O in an environment designed for either fixed length or variable length records.

STRUCTURED PROGRAMMING

IF...ELSE...ENDIF

Use these structured programming commands to get rid of those unnecessary GOTO's. Allows any number of statements for either the true or false condition.

WHILE...ENDWHILE

As with all **BASIC A +** control structures, WHILE...ENDWHILE may be nested to any level (subject to available memory).

BGET/BPUT

Provides the **BASIC A +** user with assembly level capabilities. Whole blocks of data can be quickly moved between any location and any file or device.

C/65

Another FIRST from OSS!! C/65 is the FIRST commercially available C compiler for both the Apple and Atari which actually produces assembly language output. (We know of other similar C compilers, but they all produce a "p-code" output which then must be interpreted.) C/65 is based on the "Small C" compiler, as published in *Dr. Dobbs Journal*, but it has been much restructured to enable it to run on and produce code for our familiar 6502 based machines.

C/65 is a usable subset of the powerful C language: it supports INTEGERS and CHARACTERS, arrays thereof, and pointers thereto. Naturally, it also features full recursion, easy assembler interface, #INCLUDE, and a non-macro version of #DEFINE. AUTOMATIC, global and EXTERNAL variables are also available; and, even though neither static nor initialized variables are directly supported, one may easily include assembly language code which effects the desired results. And the language DOES work: we have used it to write several new OS/A + commands.

Now be one of the FIRST to be using the ever popular C programming language on your Atari or Apple computer.

C/65 requires MAC/65 or an equivalent assembler.

tiny-c

As part of our continuing effort to support the Atari user community, we have — through special arrangement with tiny-c Associates — developed **tiny-c** for your use. **tiny-c** provides an easy-to-use, easy-to-modify programming environment for any suitably equipped Atari computer. **tiny-c** is an interactive, INTERPRETIVE language that implements a very usable subset of the extremely power C language.

OSS now then proudly presents **tiny-c** for your Atari Home Computer. **tiny-c** includes the Portable Program Preparation System for ease of software development. Since FULL source is included, the user may customize the system as desired. The comprehensive library which is included allows easy access to all features of OS/A +.

tiny-c is undoubtedly the easiest introduction ever to the world of structured languages. **tiny-c** could very appropriately be used as a language for programming instruction, as as first step up from BASIC, or as an experimenter's tool.

The OSS **tiny-c** package includes a comprehensive user's guide, which serves both as an introduction to structured programming and as a **tiny-c** reference manual.

SpeedRead +

Busy People • Executives • Secretaries
Teachers • Students • EVERYONE

Yes, everyone can benefit from **SpeedRead +**.

For the first time anywhere, the full power of the personal computer is being used to help you read faster and better.

SpeedRead + goes far beyond mere mechanical devices are capable of: **SpeedRead +** functions at your reading speed, from now to thousands of words per minute. **SpeedRead +** can help you overcome such reading problems as vocalization, single word stops, sub vocalization, and more.

SpeedRead + offers four modes designed to speed your progress in reading for speed, enjoyment, comprehension, and retention:

Single Phasefor eye recognition of words and phrases
Double Phasefor eye movement, focus, and timing
Random Phasefor peripheral vision expansion
Column Phasescanning paragraphs with one stop per line

In all four modes, **SpeedRead +** offers you additional features which help you train your eyes and brain to function as the incredible precision machine they were meant to be:

- 5 selectable phrase widths from one word to one line
- 200 selectable speeds from 5 to over 2000 words/minute
- Easy control of selection usually single keystroke
- Change speeds while reading via keystroke or joystick

SpeedRead + comes complete with classic short stories for your reading practice and pleasure. But **SpeedRead +** was designed for ease of use and expansion, so use any appropriately named text file: a story, stock market report, the news, etc.

Naturally, the **SpeedRead +** package includes a comprehensive multi-level user guide, complete with summary instructions, detailed explanations of each choice, and explanations of slow reader problems and how to correct them with appropriate **SpeedRead +** exercises. Also included is a simple multiple choice examination program, for use with user supplied quizzes.

SpeedRead + for Atari Home Computers requires 16K and a disk drive. **SpeedRead +** for the Apple II uses 48K and a disk drive. Inquire for details on other versions.

FIRST AND FINEST in Systems Software for Apple and Atari Computers

FIRST we delivered Atari's Assembler Editor (the cartridge).

Then we produced our enhanced EASMD.

NOW OSS is introducing the finest integrated assembly language development system yet.

Naturally, **MAC/65** is upward compatible with both EASMD and the Atari Cartridge. And of course, the object code output is also compatible with OS/A+, Atari DOS and/or Apple DOS, as appropriate.

MAC/65 **\$80.00**

OS/A +

OS/A + is the **FIRST AND FINEST** operating system for both the Apple II and Atari computers and features a keyboard driven, easy-to-use command processor. In addition to several simple resident commands, OS/A+ allows logical and readable requests for even the most sophisticated utility commands

AND NOW OS/A + is included as a part of every standard OSS language package.

BUG/65

Introducing **BUG/65**, an extremely useful and powerful debugger. **BUG/65** includes all the traditional debugging operations: change memory, display memory, disassembly, instant assembly and more. Also included is a break point capability, single step and trace modes.

BUG/65 is included **FREE** with every **MAC/65** package.

BUG/65 **\$34.95**

OPTIMIZED SYSTEMS SOFTWARE, INC.

ABOUT

If you own an Apple or Atari computer, you're already acquainted with us. The programmers of OSS Inc., produced the original Apple DOS and all versions of Atari BASIC, Atari FMS and the Atari Assembler/Editor.

Optimized Systems Software, Inc., hopes to meet your needs now and in the future. We are dedicated to serious products that span the gap between microcomputers from various manufacturers, and we welcome your suggestions, products and ideas.

BASIC A +

"FROM THE AUTHORS OF ATARI BASIC ..."

BASIC A + is the only logical upgrade available to the Atari BASIC programmer. While retaining all the features which make Atari BASIC so easy to use, we've also given **BASIC A +** features that place it at the forefront of modern interpretive languages: structured programming, superior input/output, helpful programming aids, and even a very comprehensive **PRINT USING** command. And, exclusively for the Atari computer, an almost unbelievable array of **PLAYER/MISSILE GRAPHICS** commands an functions.

BASIC A + **\$80.00**

C/65

The **FIRST** native code C compiler ever produced for Atari and Apple computers.

C/65 supports a very usable subset of the extremely powerful and popular C language. Just as C is used by the most sophisticated programmers from the professional and academic communities, so shall **C/65** prove to be a powerful and much-needed tool for 6502 software developers.

C/65 **\$80.00**

SpeedRead +

The first and still finest speed reading tutor designed for you to use on your computer is available only from OSS.

SpeedRead + uses time-proven techniques to train you to instantly recognize words and phrases, exercises your peripheral vision, improves your eye movement and timing, and generally works with you at your pace... now and in the future.

SpeedRead + **\$59.95**

TINY C

As a product of Tiny-C Associates, **tiny-c** was the first structured language interpreter for microcomputers. Now OSS brings this innovative interpretive language to your home computer. While not having the speed of our C/65 compiler, **tiny-c** is an excellent choice for the programming student who is ready to begin learning the valuable techniques of structured programming.

tiny-c **\$99.95**





For Atari Computers
THE WRITER'S
TOOL™

*with Frank Goughlin
and John
Gibson*

QSS

© 1984 Madison Micro & OSS, Inc.

For Atari Computers
THE WRITER'S
TOOL™ SINGLE

1.5/36 Density

2 1/2" Diskettes
Z Z5

Master Copy 1

QSS

© 1984 Madison Micro & OSS, Inc.

*BXE WORK
INCLUDING FORMATTED
INPUT*

MASTER DISKETTE

DOS XL

Optimized by

BASIC XE
EXTENSION DISK
(For Atari Computers)

Version 4.11 **QSS**

© 1984 Madison Micro & OSS, Inc.

For Atari Computers
THE BASIC XL
TOOLKIT™

QSS

BASIC XE
EXTENSION DISK
(For Atari Computers)

LISTINGS

QSS

INDUS
SYSTEMS

